Failure Distance-based Simulation of Repairable Fault-Tolerant Systems

Juan A. Carrasco
Departament d'Enginyeria Electrònica
Universitat Politècnica de Catalunya
Diagonal 647, plta. 9
08028 Barcelona, Spain
juan.a.carrasco@upc.edu

Except for formatting details and tiny corrections, this version matches exactly the version published with the same title and authors in *Computer Performance Evaluation Modeling Techniques and Tools*, G. Balbo and G. Srazzi (eds), Elsevier, 1992, pp. 351–365

Abstract

This paper presents a new importance sampling scheme called failure biasing for the efficient simulation of Markovian models of repairable fault-tolerant systems. The new scheme enriches the failure biasing scheme previously proposed by exploiting the concept of failure distance. This results in a much more efficient simulation with speedups over failure biasing of orders of magnitude in typical cases. The paper also discusses the efficient implementation of the new importance sampling scheme and presents a practical method for the optimization of the biasing parameters.

NOTE FROM THE AUTHOR: The method proposed in the paper for the optimization of the biasing parameters introduces correlation, making the estimates invalid. A more recent paper, J. A. Carrasco, "Failure Transition Distance-Based Importance Sampling Schemes for the Simulation of Repairable Fault-Tolerant Computer Systems, *IEEE Trans. on Reliability*, vol. 55, no. 2, June 2006, pp. 207–236, minor corrections in *IEEE Trans. on Reliability*, vol. 56, no. 2, June 2007. p. 360, presents two slightly modified biasing schemes which can be proved to be more efficient for balanced systems than failure and balanced failure biasing, and describe a correct and efficient method for the optimization of the biasing parameters. The author apologizes for the error in this paper.

1 Introduction

Availability/reliability metrics are appropriate for the evaluation of reapairable fault-tolerant systens which from the user's point of view can be seen as either operational or down. Important metrics of this type are the steady-state availability, the availability, the interval availability, the mean time to failue (MTTF), and the reliability. For the computation pf these metrics the system can be viewed as made up of instances of component types which change their state as a result of failure and repair processes. Under the assumption of exponential failure and repair time distributions, homogeneous continuous-time Markov chains (CTMCs) area a powerful modeling tool, well suited to capture all sort of dependencies which realistic models have to consider.

The main problem of CTMCs (and in general of any type of stochastic state-level models) is the exponential growth of their size with the number of component types of the system. Simulation is an approach which by nature is not limited by the size of the model, but, for repairable fault-tolerant systems, the values of the metrics which are really of interest (i.e., the steady-state unavailability, since the steady-state availability is usually very close to 1) result from contributions of rare paths and direct Monte Carlo simulation is unfeasible. Two types of techniques have been proposed to speed up direct Monte Carlo simulation. Inmportance sampling techniques exploit heuristic knowledge about the model to modify the sampling distributions so that the rare contributing paths be sampled more often. Failure biasing and forced transition are two such techniques which were initially proposed in the context of the nuclear domain [1, 2], and have been recenty further developed and applied with success to the simulation of models of fault-tolerant computer systems [3]–[5]. Estimator decomposition techniques exploit heuristic knowledge abot the models to formulate the metric of interest in terms of lower-level metrics which can be estimated more efficiently. Such techniques have been recently used for the estimation of the steady-state availability [6] and the MTTF [7], as well as the availability, the interval availability, and the reliability [8].

This papr presents a new importance sampling scheme called *failure distance biasing* which enriches the failure biasing scheme previously poposed by exploiting the concept of failure distance. As the examples presented in Section 7 illustrate, the new scheme can achieve speedups over failue biasing of orders of magnitude in typical cases. The paper also discusses the efficient computation of failure distances, which are required by the scheme, and gives a practical method for the optimization of the biasing parameters. The rest of the paper is organized as follows. Section 2 describes the type of models under consideration. Section 3 contains a brief review of availability/reliability simulation. Section 4 describes the new importance sampling scheme. Section 5 presents efficient techniques for the computation of failure distances, as required by the proposed scheme. Section 6 describes a practical and efficient method for the optimization of the biasing parameters of the scheme. Experimental results are presented in Section 7 illustrating both the speedups over failure biasing and the efficiency of the techniques proposed for the computation of failure distances. Section 8 concludes the paper and outlines future reserach directions.

2 Type of Models

In the models under consideration the system is viewed as made up of instances of component types, the instances of the same component type being completely indistinguishable. Components are either unfailed or failed and, in general, can be failed in several modes. The system is operational or down as determined by a coherent structure function [9] of the unfailed/failed state of the components of the system. Without loss of generality, we assume that the structure function is represented by a fault-tree consisting of and, or gates. The fault-tree can have in general fanout and its inputs have associated atoms of the form $t[k]$ with the semantics "at least k components of type t are failed". The output of the fault-tree evaluates to true if and only if the system is operational. An alternative representation, which sometimes is more convenient, is provided by an operation-tree whose output evaluates to true if and only if the system is operational and in which the atoms $t[k]$ have the semantics "at least k components of type t are unfailex". Both representations are equivalent, in the sense that it is possible to transform an operation-tree into a fault-tree representing the same structure function and viceversa by transforming and gates into or gates and or gates into and gates and replacing each atom $t[k]$ by $t[n+1-k]$, where n is the number of instances of type t in the system.

The state of the system changes as a result of failure and repair processes with constant but, possibly, state-dependent rates. Failure processes are associated with components, but the failure of a component can in general be propagated to others. Components without failure processes associated with them are called *non-failing* and provide a very general framework for the modeling of lack of coverage. For instance, system failures due to lack of coverage can be modelled by introducing a non-failing "recovery" component to which uncovered failures are propagated and requiring the "recovery" component to be unfailed for the system to be operational. The repair of the "recovery" component would model a system restart. It is also possible to model in this way covergae failures taking down only part of the system.

We assume that all component types are repairable and that failed components are immediately considerd for repair according to a "static" repair policy, which only takes into account the current state of the components. Under this hypothesis, the behavior of the system can be modelled by a finite ergodic CTMC $X(t)$, whose states can be described by the number of components of each type in each component state and whose transitions are associated to either failure or repair processes. The state with all components unfailed will be denoted by u. All states except u have outgoing failure and repair transitions. The state u has only outgoing failure transitions.

3 Availability/Reliability Markovian Simulation

It has been recently showed [6]–[8] that availability/reliability simulation of repairable fault-tolerant systems can be speeded up by using estimator decomposition techniques. The idea is to formulate the metric of interest in terms of lower-level metruics which can be expressed as the expected value

of a path function over the regenerative behavior of $X(t)$ around u, use independent simulation streams to obtain estiamtes for the low-level metrics and combine these estimates to achieve the estimate for the desired metric. For instance, the steady-state unavailability can be simulated based on the formulation:

$$ua = \frac{\tau_{uu}^D}{\tau_{uu}}\,, \tag{1}$$

where τ_{uu}^D is the mean time spent by $X(t)$ during a regenerative cycle in the subset of down states D and τ_{uu} is the mean regenerative cycle duration. Let $\Pi(t)$ be the transient CTMC with initial state u and absorbing state a capturing the regenerative behavior of $X(t)$. Let r be a path to absorption of $\Pi(t)$ and denote by L_r the length (number of transitions) of r, by $x_i(r)$ the ith visited state ($x_o(r) = u$), and by h_x the mean holding time in x. τ_{uu}^D is the expected value of the path function $Z_r = \sum_{0\le i\le L_r-1, x_i(r)\in D} h_{x_i(r)}$ and τ_{uu} is the expected value of the path function $Z_r = \sum_{0\le i\le L_r-1} h_{x_i(r)}$. We next present a brief review of importance sampling theory [10] in the xontext of path simulation of $\Pi(t)$.

Let R be the set of paths to absorption of $\Pi(t)$ and denote by q_{ij} the jump probability from state i to state j. The probability of a path $r \in R$ is given by:

$$P(r) = \prod_{i=0}^{L_r-1} q_{x_i(r),x_{i+1}(r)}\,.$$

Let ρ be the random variable "path to absorption followed by $\Pi(t)$" and Z_r a path function. We want to estimate:

$$E[Z_\rho] = \sum_{r\in R} Z_r P(r)\,.$$

Assume we want to achieve a given confidence interval of given relative width with respect to $E[Z_\rho]$. The number of paths, M, which have to be sampled in direct Monte Carlo simulation is proportional to $\sigma^2(Z_\rho)/E[Z_\rho]^2$, which, using $\sigma^2(Z_\rho) = E[Z_\rho^2] - E[Z_\rho]^2$, can be expressed as:

$$\frac{\sigma^2(Z_\rho)}{E[Z_\rho]^2} = \sum_{r\in R}\left(\frac{Z_r P(r)}{E[Z_\rho]}\right)^2 \frac{1}{P(r)} - 1\,, \tag{2}$$

where $Z_r P(r)/E[Z_\rho]$ can be interpreted as the relative contribution of r to $E[Z_\rho]$. Therefore, (2) says that M will be large if paths with significant contributions have small probabilities. In order to reduce the simulation effort we can sample the paths with biased probabilities $P^*(r)$ ($P^*(r) \neq 0$ whenever $Z_r P(r) \neq 0$) and take the sample mean of the path function $Z_r^* = Z_r \Lambda^*(r)$, where the likelihood ratio $\Lambda^*(r) = P(r)/P^*(r)$ is introduced so that $E[Z_{\rho^*}^*] = E[Z_\rho]$. The goal is to choose $P^*(r)$ so that the variance $\sigma^2(Z_{\rho^*}^*)$ of the new path function be substantially smaller than $\sigma^2(Z_\rho)$. It is easy to show that $\sigma^2(Z_{\rho^*}^*) = 0$ when $P^*(r) = Z_r P(r)/E[Z_\rho]$ and importance sampling theory suggests to sample paths with probabilities $P^*(r)$ as close as possibel to their relative contributions to the metric. When, as in this paper, $\Pi(t)$ si biased by modifying the jump probabilities the likelihhod ratio can be expressed as:

$$\Lambda^*(r) = \frac{\prod_{i=0}^{L_r-1} q_{x_i(r),x_{i+1}(r)}}{\prod_{i=0}^{L_r-1} q^*_{x_i(r),x_{i+1}(r)}}\,. \tag{3}$$

Repair rates are usually several orders of magnitude higher than failure rates. This has two implications for the models under consideration. First, the probability of following a failure transition from a state $x \neq u$ is typically very small. Second, the mean holding time h_x for the states $x \neq u$ is typically much smaller than h_u. Consider now the simulation of the steady-state unavilability based on (1). It follows from the previous observations that the value Z_r for the path function associated to τ_{uu} (which is the mean duration of the path) is $\approx h_u$ for all paths with signficant probabilities. This implies that these paths have relative contributions to $\tau_{uu} \approx P(r)$ and, according to importance sampling theory, τ_{uu} is estiamted very efficiently by direct Monte Carlo simulation. On the other hand, direct Monte Carlo simulation would be highly inefficient for the estimation of τ_{uu}^D, since only paths of $\Pi(t)$ entering D have non-null contributions to this metric and typically these paths have globally a small probability. A similar scenario arises in the simulation methods proposed in [7] for the MTTF and in the methods proposed in [8] for the availability, interval availability and reliability. Simulation of the badly-behaved low-level metric can be speeded up by using an importance sampling scheme in which paths entering D are sampled with high probability. Failure biasing is such an scheme which was proposed in [1, 2] and borrowed in [3, 5, 6, 7] for the simulation of the type of models considered in this paper. The scheme biases the jump probabilities from the states $x \neq u$ so that the probability of following a failure transition is $FBIAS$ and the probability of following a repair transition is $1 - FBIAS$. The simulation effort (number of events) is minimized by choosing a value for $FBIAS$ which typically is close to 0.5.

4 Failure Distance Biasing

Although failure biasing succeeds in sampling with a substantial global probability the contributing paths of the badly-bhaved low-level metric, it does not fully exploit their heuristically clear importance ranking. Consider, for instance, a system, with m component types and two components of each type, which is operational if at least one component of each type is unfailed, and assume that all components fail independently (no failure propagation) with the same rate. The probabilities of the paths of $\Pi(t)$ decrease in general very fast with their length and, according to importance sampling theory, shorter contributing paths should be sampled more often than longer contributing paths. In the example, after sampling from u the transition associated with the failure of a component, we should sample the transition associated with the failure of the other component of the same type with higher probability than the transitions associated with the failure of the other components. But using the failure biasing technique all these transitions would be sampled with the same probability.

The importance sampling scheme proposed in this paper is based on the concept of *failure distance*. The failure distance from a down state is 0. The failure distance from an operational state x is the minimum number of *failing* components whose failure in x would take the system down. Let $t = (x, y)$ denote a failure transition from x to y and let $d(x)$ denote the failure distance from state x. We say that t is *non-dominant* if $d(y) = d(x)$, *dominant* if $d(y) < d(x)$, and *critical* if $d(y) < d(x) - 1$. Critical failure transitions are always associated to failure processes involving several components. The *criticality* of the failure transition t is defined as $c(t) = d(x) - d(y)$.

As in failure biasing, we turn our achem off when a down state is hit. When biasing is on the jump probabilities from a state are modified in a process which can be described as done in steps. At each step, a subset of transitions is split into two subsets, which are biased in relation to one another, and one of the subsets is passed to the next step. If one of the subsets is empty, the step is skipped. As in failure biasing, the first step assigns a probability $FBIAS$ to the set of failure transitions, which is passed to the next step, and a probability $1 - FBIAS$ to the set of repair transitions. In the next step, the set of dominant failure transitions is assigned a probability $DBIAS$ in relation to the current set and passed to the next step, and the set of non-dominant failure transitions is assigned a relative probability $1 - DBIAS$. The thirs step is repeated while the transitions in the current set have different criticalities and assigns the relative probability $1 - CBIAS$ to the set of transitions with the smallest criticality and the relative probability $CBIAS$ to the complmentary set, which is the one considered for the next application of the biasing step. Assume, for instance, that the current state has repair transitions, non-critical dominant failure transitions, and critical falire transitions of criticalities 2 and 3. In failure distance biasing, these subsets of transitions are sampled with, respectively, the probabilities $1 - FBIAS$, $FBIAS(1 - CBIAS)$, $FBIAS\ CBIAS(1 - CBIAS$, and $FBIAS\ CBIAS^2$.

The biasing parameters $DBIAS$ and $CBIAS$ control the focus of the sampling to the shorter paths entering D. The use of an independent biasing parameter to deal with critical failure transitions is convenient since the actual importance of the paths containing this type of transitions depend on the values of the "covergae" parameters of the model. Note also that non-failing components are not taken into account in the definition of the failure distance. This is done so that the presence of "recovery" compoments (see Section 2) do not affect the heuristics behind the biasing scheme. Consider, for instance, a system with a "recovery" component which has to be unfailed for the system to be operational. If non-failing components were taken into account, the failure distance from all the operational states would be 1 and all failure transitions not taking the system down would be biased equally.

A variant of the failure biasing scheme which is somhow related to the scheme proposed here is described in [5]. Under that scheme, failures of component types which have already some instance failed are biased independently of the others. This scheme does not take into account that (as illustrated by the example presented in Section 7) redundancy can be provided between components of different types. In addition, the scheme will bias equally all failure transitions from the state u.

5 Computation of Failure Distances

Application of the failure distance biasing scheme requires the computation of the failure distance from the current state and the states reached from it by failure transitions. These distances can be computed using the minimal cuts of the structure function of the model. Since this function is defined in terms of instances of component types, a minimal cut m is specified by a set of component types $tinmc(m)$, and, for each component type t in the set, by the number of instances $inmc(t, m)$

of t in m. Denote by NFC the set of non-failing component types and by $failed(t,x)$ the number of instances of type t which are failed in the state x. Let $U(k)$ be the function returning k if $k > 0$ and 0 otherwise. We define the distance $dtomc(x,m)$ from x to the minimal cut m as ∞ if $inmc(t,m) > failed(t,x)$ for some $t \in tinmc(m) \cap NFC$, and as

$$\sum_{t \in tinmc(m)} U(inmc(t.m) - failed(t,x))$$

otherwise. Denote by MC the set of minimal cuts. We have:

$$d(x) = \min_{m \in MC} dtomc(x,m)\,. \tag{4}$$

Let f be a failure event, i.e., a set of components which can fail simultaneously and denote by $ad(x,f)$ the failure distance from a state reached from x by a failure transition involving the components in f. Let AMC_f be the set of minimal cuts of the structure function obtained from the structure function of the model by failing the component instances in f. AMC_f can be obtained by considering the minimal cuts with instances of the component types in f and removing from them as many component instances of f as possible. If f includes only one component instance the cuts thus obtained are guaranteed to be minmal; otherwise, the set has to be reduced. It is easy to show that:

$$ad(x,f) = \min\{d(x), \min_{m \in AMC_f} dtomc(x,m)\}\,. \tag{5}$$

Using (4), (5), the criticality of the failure transitions can be computed if a priority queue yielding $\min_{m \in MC} dtomc(x,m)$ and a priority queue yielding $\min_{m \in AMC_f} dtomc(x,m)$ for each failure event f of the model are updated as the path is sampled.

The number of minimal cuts can be large when the system has many component types and bookkeepping the distances to all of them can be expensive. The number of cut "touches" can however be reduced significantly by exploiting the following observation. Consider, for instance, the bookkeepping of $d(x) = \min_{m \in MC} dtomc(x,m)$ and assume that an upper bound ub for $d(x)$ is known. Denote by $o(m)$ the order (number of components) of the minimal cut m and by $n(x,m)$ the number of components in m which are failed in x. Since $dtomc(x,m) \geq o(m) - n(x,m)$ (it will be $>$ if m has unfailed instances of non-failing component types) m does not need to be considered for the computation of the minimum distance if $o(m) - n(x,m) \geq ub$.

The bookkeepping of the distances to the minimal cuts in MC and the distances to the minimal cuts in the sets AMC_f is done independently. In a given state x, only he minimal cuts m with $o(m) \leq K$, an for each order $k \leq K$ only those with $n(x,m) \geq R(k)$ have their distances updated. The remaing cuts have their distances in the priority queues set to $dtomc(u,m) \geq dtomc(x,m)$. The values of K and $R(k)$ are selected so that the minimal cuts m whose distance is "cleaned" are guaranteed to have a distance non smaller than a known upper bound for, respectively, $d(x)$ and $\max_f ad(x,f)$. In addition, $R(k)$ is not allowed to take a value greater than the parameter R.

Let s be the state visited before x and denote by $n(x)$ the number of components failed in x. The bookkeepping after a failure transition associated to a failure event f is done as follows. The

distances to the minimal cuts in MC are not updated since $d(x) = ad(s, f)$, which is known. To obtain the after failure distances $ad(x, f)$ we update the distances to the minimal cuts in the sets AMC_f as follows. First, we set the global upper bound ub to $d(x)$. Then, we consider the minimal cuts in increasing order k while $k - n(x) < ub$, and, for each order k, we compute the new value of $R(k)$ as $\max\{1, \min\{k - ub + 1, R\}\}$ and update the distances to the minimal cuts of order k so that only those with $R(k)$ or more failed components in x have their distances updated and the remaining minimal cuts of order k have their distances "cleaned". After processing the minimal cuts of a given order the upper bound ub is updated considering the values at the top of the priority queues associated to the failure events, and at the end of the while loop, we clean the cuts of order higher than the maximum processed which had their distances updated. The values obtained at the top of the priority queues are guaranteed to be the minimum distance to the after minimal cuts associated to each failure event.

The bookkeepping after a repair transition r is done as follows. In order to compute $d(x)$ we first clean the distances to the "touched" minimal cuts and perform an updating process similar to the one described before. Let n_r be the number of components repaired in r. The upper bound for $d(x)$ is initially set to $\min\{d(s) + n_r, d(u)\}$ if r involves only failing component types, and to $d(u)$ otherwise. To obtain the after failure distances $ad(x, f)$ we first update the distances to the "touched" minimal cuts in the sets AMC_f to consider the component instances repaired in r and then follow the same upodating procedure as before, setting initially the global upper bound to $\min\{\max_f ad(s, f) + n_r, \max_f top(f), d(x)\}$ ($top(f)$ denotes the value at the top of the priority queue associated to f) if r involves only failing component types, and to $\min\{\max_f top(f), d(x)\}$ otherwise.

In order to drive the updating process we use minimal cut selectors. A selector is a distinguished combination of component instances which is part of some minimal cut, and has associated a list linking the minimal cuts including the selector. Thus, access to the minimal cuts with at least n failed instances ($n \leq R$) can be done by following the lists associated to the minimal cut selectors of order n having all their component instances failed.

We also tried a more ellaborated variant in which the bookkeepping of the minimal cuts in each set AMC_f was controlled independently and found that the overhad was excesive. Usually, the optimized values for the biasing parameters are such that the sampling is strongly focussed to paths including mostly failure transitions reducing the failure distance and the bookkeepping strategy will almost restrict distance updates to the cuts of minimum order with increasing number of failed components.

6 Biasing Optimization

In this section we propose an adaptive optimization scheme for the minimization of the required simulation effort which we have found robust and efficient. In the adaptive optimization scheme the simulation stream is split into substreams and the biasing parameter values are optimized at the end

of each substream. The lengths of the substreams are chosen so that the total simulation length is approximately doubled after each of them.

The number of events which is required to achieve a given confidence intervak is proportional to:

$$\mu^* = \sigma^2(Z^*_{\rho^*})E[L_{\rho^*}]\,. \tag{6}$$

A general minimization method is provided by the likelihood ratio gradient theory [11]. However, a simbolic estimator for μ^* as a function of the biasing parameters can be obtained as the simulation progresses and used with advantage.

Let $P'(r)$ and $\Lambda'(r)$ be the actua l sampling probabilities and likelihood ratios (corresponding to the current values of the biasing parameters), and $P^*(r)$ and $\Lambda^*(r)$ the generic values. Considering that $\sigma^2(Z^*_{\rho^*}) = E[(Z^*_{\rho^*})^2] - E[Z^*_{\rho^*}]^2$ and $E[Z^*_{\rho^*}] = E[Z_\rho]$, independent of the biasing parameters, we can write (6):

$$\mu^* = \left(\sigma^2(Z'_{\rho'}) + E[(Z^*_{\rho^*})^2] - E[(Z'_{\rho'})^2]\right) E[L_{\rho^*}]\,. \tag{7}$$

Using $\Lambda^*(r)P^*(r) = \Lambda'(r)P'(r) = P(r)$, we can formulate $E[(Z^*_{\rho^*})^2]$ and $E[L_{\rho^*}]$ as:

$$E[(Z^*_{\rho^*})^2] = \sum_{r\in R} Z_r^2\Lambda^*(r)^2P^*(r) = \sum_{r\in R} Z_r^2\Lambda^*(r)\Lambda'(r)P'(r)$$

$$E[L_{\rho^*}] = \sum_{r\in R} L_rP^*(r) = \sum_{r\in R} L_r\frac{\Lambda'(r)}{\Lambda^*(r)}P'(r)\,,$$

which tells us that $E[(Z^*_{\rho^*})^2]$ and $E[L_{\rho^*}]$ can be estimated by the sample means of, respectively, the path functions:

$$Y_r = Z_r^2\Lambda^*(r)\Lambda'(r)$$
$$W_r = L_r\frac{\Lambda'(r)}{\Lambda^*(r)}\,.$$

The generic biasing jump probability of the ith transition of a path r can be expressed in terms of the biasing parameters as:

$$q^*_{x_{i-1}(r),x_i(r)} = \frac{q_{x_{i-1}(r),x_i(r)}}{Q(r,i)} \times FBIAS^{n_1(r,i)}(1-FBIAS)^{n_2(r,i)}\cdots(1-CBIAS)^{n_6(r,i)}\,,$$

where $Q(r,i)$ is the unbiased probability of the class (repair or failure with given criticality) to which the transition belongs and $n_j(r,i)$ are integers ≥ 0. Denoting by $B(r)$ the set of transitions of path r coming from a state in which biasing is active, and using (3) we get:

$$\Lambda^*(r) = \frac{\prod\limits_{i\in B(r)} Q(r,i)}{FBIAS^{n_1(r)}(1-FBIAS)^{n_2(r)}\cdots(1-CBIAS)^{n_6(r)}}\,,$$

where $n_j(r) = \sum_{i\in B(r)} n_j(r,i) \geq 0$. Then the sample means $\overline{Y}_r$ and $\overline{W}_r$ belong, to, respectively, the class of functions:

$$f(x_1,x_2,\ldots,x_n) = \sum_{k=1}^{p} a_k \prod_{i=1}^{n} \frac{1}{x_i^{n_i(k)}(1-x_i)^{n'_i(k)}}$$

$$g(x_1, x_2, \ldots, x_n) = \sum_{k=1}^{l} b_k \prod_{i=1}^{n} x_i^{n_i(k)} (1 - x_i)^{n'_i(k)} ,$$

with $a_k, b_k > 0$, and $n_i(k)$, $n'_i(k)$ integers ≥ 0. Symbolic expressions for $\overline{Y}_r$ and $\overline{W}_r$ can be obtained as paths are sampled by accumulating the factors a_k, b_k. The symbolic estimator for μ^* is finally obtained by using (7), with the variance $\sigma^2(Z'_{\rho'})$ estimated by the sample variance collected during the last substream and the remaining quantities by using the symbolic estimators for $E[(Z^*_{\rho^*})^2]$ and $E[L_{\rho^*}]$.

The symbolic estimator for μ^* may have several local minima. Global minimization procedures are too expensive for our context and are not always guaranteed to return the global minimum. However, we have found that the variance $\sigma^2(Z^*_{\rho^*})$ is much more sensitive to the biasing parameters than $E[L_{\rho^*}]$ and a minimization of $\sigma^2(Z^*_{\rho^*})$ yields optimized biasing parameter values close to those resulting from the minimization of μ^*. Taking this into account we first minimize $E[(Z^*_{\rho^*})^2]$ (which is equivalent to the minimization of the variance) and starting from that point we make a local minimization of μ^*. It is proved in [12] that the class of functions $f(x_1, x_2, \ldots, x_n)$ to which the symbolic estimator for $E[(Z^*_{\rho^*})^2]$ belongs are convex in the domain $]0, 1[^n$, and then a local minimization algorithm is enough [13] to minimize $E[(Z^*_{\rho^*})^2]$.

Using the adaptive biasing optimization scheme, we are in fact sampling a different random variable $Z^*_{\rho^*_i}$ at each substream i. The final estimate for $E[Z_\rho]$ is computed by weighting optimally the sample means according to estimates for their variances, which are obtained as follows. Let M_i be the number of paths of the ith substream, s_i^2 the sample variance for the ith substream, and S_i the current estimate for $E[(Z^*_{\rho^+})^2]$ for the values of the biasing parameters used in the ith substream. After the nth substream we estimate the variance of the sample mean of the ith substream by:

$$\widehat{\sigma}_i^2 = \frac{1}{M_i}(s_n^2 + S_i - S_n) .$$

This procedure was selected after trying the use of the sample variances s_i^2, which was found dangereous when the model is "hard". The reason is that in such cases the sample variances tend to be optimistic for the first substreams and undue weights are assigned to the poor estimates obtained in those substreams.

7 Experimental Analysis

The results presented in this section were obtained using a prototype software package which implements the simulation methods described in [6]–[8] under both failure biasing and failure distance biasing. The interface required by the simulator includes one function returning the failure and repair processes which are active in a given state (expressed in terms of action/response pairs), two functions returning the rates and probabilities associated to actions and responses, another function returning the next state given the current state and an action/response pair, and another function determining whether the system is operational or down in a given state. These functions were obtained using the model specification preprocessor included in METFAC [14].

In our implementations of the methods we turn off the biasing schemes after a given number $MAXREP$ of repair transitions are sampled. This ensures that the variance of the estimator is finite and has little effect on the biasing scheme if a large enough value for $MAXREP$ is chosen. We have found $MAXREP = 2$ to be an appropriate choice. Other important parameters of our implementation of the biasign schemes are the length of the first "biased" substream $MINBEVENTS$, the initial values for the biasing parameters $IFBIAS$, $IDBIAS$, and $ICBIAS$, and $PINT$. The values of the biasing parameters are restricted to the interval $[PINT, 1 - PINT]$ to prevent the biasing parameters from getting too close to 0 or 1, since this would delay the reaction capability of the scheme against a premature minimization. The choixe $MINBEVENTS = 500$, $IFBIAS = 0.8$, $IDBIAS = 0.7$, $ICBIAS = 0.2$ for failure distance biasing, $IFBIAS = 0.5$ for failure biasing, and $PINT = 0.05$ have given good results in all tests we have run.

Events are optimally allocated between the simulation streams used for the low-level metrics. This typically results in allocating almost all the events to the stream used for the estimation of the badly-behaved low-level metric which is sampled with biasing.

The biasing schemes will be compared using a non-trivial large example for which simulation would be a competitive approach. The example is the fault-tolerant data processing system whose architecture is shown in Figure 1. A dual configuration of data processing units (DPUs) command control subsystems located at remote sites. Each control subsystem comprises two redundant control units (CUs) working in hot-standby redundancy. The system can be accessed through two redundant front-ends connected to the DPUs. The DPUs and CUs communicate using a redundant local area network (LAN) to which each DPU and each CU has access through dedicated communication processors (CPs). All components fail with constant rates λ_{FE}, λ_{DPU}, λ_{CU}, λ_{CP}, and λ_L, respectively. Two failed modes are consider for the DPUs: "soft" and "hard". The first mode occurs with probability α and can be recovered by an operator restart; the second mode occurs with probability $1 - \alpha$ and requires hardware repair. Coverage is assumed perfect for all faults except those of the DPUs, which take the system own with a probability $1 - C$. Lack of coverage is modelled by propagating the failure of Sone DPU to the other DPU. There are three repair teams. The first repairs LANs and CPs, with preemptive prority given to LANs. The second repairs FEs, DPUs and CUs in "hard" failed mode, with preemptive priority given first to DPUs, next to FEs, and last to CUs. The third makes DPU restarts. Each team includes only one repairman. Failed components with the same repair priority are taken at random for repair. The repair rates are denoted bym respectively, μ_{FE}, μ_{DPUh}, μ_{DPUs}, μ_{CU}, μ_{CP}, and μ_{rL}.

The system is considered operational if one unfailed DPU can communicate with at least one unfailed CU of each control subsystem. Different LANs can be used for communication between the active DPU and the active CU of each control subsystem, but the communication has to be direct, i.e., involving only one CP of each unit and one LAN. The front-ends can be conceptualized as being instances of the same component type. However, the interconnection relationships make it mandatory to consider all the other components as unique representatives of different component types. The resulting CTMC has about 4.6×10^{11} states, whch clearly precludes both generation and numerical solution of the state-level model.

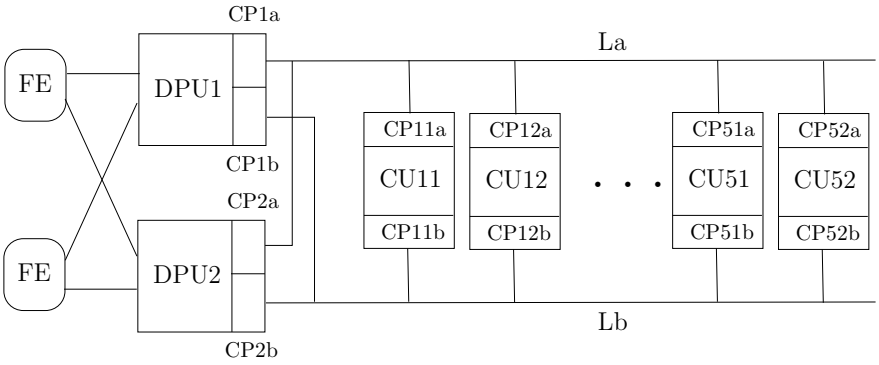

Figure 1: Fault-tolerant data processing system.

Table 1: Sets of model parameter values used in the tests.

case	a	b	c	d	e
λ_{FE}	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-5}
λ_{DPU}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-4}
λ_{CU}	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}	10^{-5}
λ_{L}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-5}
λ_{CP}	5×10^{-5}	5×10^{-5}	5×10^{-5}	5×10^{-5}	5×10^{-4}
α	0.9	0.9	0.9	0.9	0.9
C	0.9	0.99	0.999	0.999	1
μ_{FE}	0.5	0.5	0.5	0.05	5
μ_{DPUh}	0.5	0.5	0.5	0.05	5
μ_{DPUs}	4	4	4	0.4	40
μ_{CU}	0.5	0.5	0.5	0.05	5
μ_{L}	0.2	0.2	0.2	0.02	2
μ_{CP}	0.5	0.5	0.5	0.05	0.5

We have used several sets of model parameter values, representing different scenarios. The test sets are given in Table 1. The values for failure and repair rates chosen for cases a, b, c are meant to be typical, i.e., repair rate/failure rate ratios of two to three orders of magnitude and differences in failure rates of up to two orders of magnitude. These test sets only differ in the value chosen for C, the coverage to DPU failures. In case a, coverage failures are the dominant source of system failures, in case c resource exhaustion is the dominant source, and in case b both are important. Case d represents a situation in which repair dominance is weak (i.e., the probability of following a repair transition is not very close to 1). Case e accounts for the situations in which failure modes with a high number of failed components have important contributions.

We simulated the steady-state unavailability ua under both biasing schemes with a goal of a $99\ \%$ confidence interval of $\pm 2\ \%$ and a limit of 500,000 events for all cases. For failure distance biasing, the parameter R was set to 2. Our biasing scheme achieved significant speedups in all cases.

Table 2: Results obtained for ua under failure distance biasing (D) and failure biasing (F).

set	estimate	e_D	r_D	r_F	e_{su}	t_{su}
a	5.187×10^{-5}	9,353	0.0198	0.0285	113	95.3
b	7.627×10^{-6}	32,630	0.0197	0.0754	229	220
c	3.166×10^{-6}	79,594	0.0199	0.1057	181	162
d	2.911×10^{-4}	3.664×10^{5}	0.0196	0.0820	24.4	21.0
e	7.854×10^{-10}	5.1×10^{5}	0.0484	0.643	173	148

Table 2 shows the results. The subscripts D and F make reference to the results obtained under, respectively, failure distance biasing and failure biasing. We give the estimates obtained using failure distance biasing, the number of simulated events under failure distance biasing e_D (with failure biasing the limit was used up in all cases), the relative semiwidths of the $99\,\%$ confidence intervals (r_D, r_F), and two speedup factors: e_{su} and t_{su}, giving, respectively, the ratio between the number of events and CPU times which would be required by failure biasing and failure distance biasing to achieve the same confidence interval. These factors are computed as $e_{su} = (e_F r_F^2)/(e_D r_D^2)$ and $t_{su} = (t_F r_F^2)/(t_D r_D^2)$, where t_D, t_F and e_D, e_F are, respectively, the CPU times and numbers of events obtained in the tests. We can see that a speedup of two orders of magnitude is achieved by failure distance biasing in all cases except case d. Even in that case, the speedup is significant. It is interesting to note that the speedup is also high for case e, in which the heuristic of our biasing scheme breaks down, since some system failure modes with more components are more significant than system failure modes with fewer components. In order to realize the practical implications of these speedups, let us mention that in our machone (a SUN 3/260) the simulation for case a took 100 seconds of CPU time under our biasing method and more than one hour under failure biasing.

The proposed biasing optimization method works well: in most cases, the values chosen after the first simulation substream (500 events) are already very close to the optimal ones. However, occasionally we have found cases in which the biasing parameters only get stable after a large number of events and, since the overhead associated to the optimization scheme is low, we do not find advisable to turm it off. Figure 2 shows the values for the biasing parameters used in each estimator available at the end of the simulation. It can be noted that the optimized value of μ^* (which is proportional to the simulation effort required to achieve a given confidence interval) is about six times smaller than the value corresponding to the initial values of the biasing parameters. This illustrates the importance of the optimization of the biasing parameters.

Comparison of e_{su} and t_{su} in Table 2 reveals that the overhead per simulated event of our baising scheme over failure baising is small. By profiling the code we found out that in all cases except case e the overhead due to actual minimal cut touches was negligeable compared with the remaining overhead sources. The average number of minimal cut touches per event $\bar{t}$ was 41.3 in case e, 3.82 in case d, and about 1! in the remaining cases. The latter is noticeable considering that the model has 512 minimal cuts (8 of order 2, 48 of order 3, 96 of order 4, and 360 of order 6)

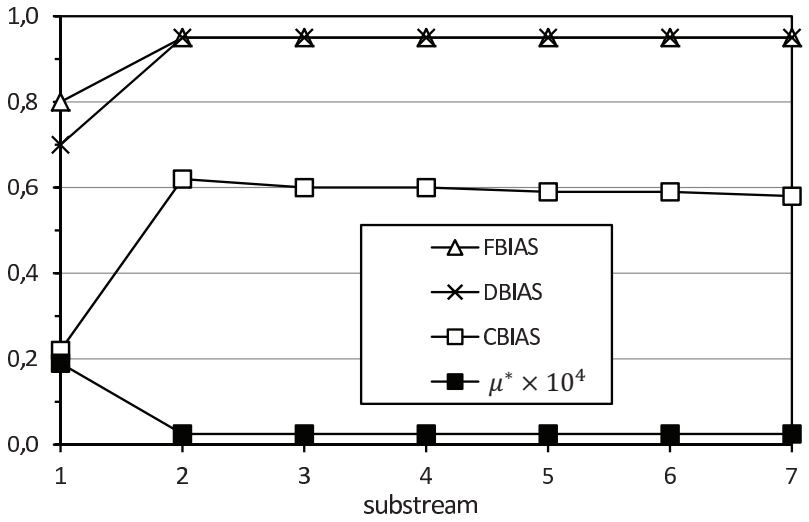

Figure 2: Behavior of the adaptive biasing parameter optimization scheme for case b.

Table 3: Average number of minimal cut touches $\overline{t}$ and average time spent per simulated event in case e for several values of R.

R	$\overline{t}$	ms/event
1	149.6	13.03
2	41.3	11.07
3	9.26	10.54
4	4.80	10.38

and 40 different failure events, and illustrates the efficiency of the techniques described in Section 5 to reduce the number of minimal cut "touches". The values of $\overline{t}$ are explained by the values of the optimized biasing parameters $FBIAS$ and $DBIAS$ for each case. The values for $FBIAS$ and $DBIAS$ are 0.95 (the maximum allowed) for cases a, b, and c, 0.84 and 0.81 for case d, and 0.95 and 0.59 for case e. As the values for $FBIAS$ and $DBIAS$ are closer to 1, the sampling is more focussed to shorte paths involving only failure transition which reduce the failure distance and minimal cuts and after minimal cuts of higher orders are less touched. Table 3 shows the values of $\overline{t}$ and the average time spent per simulated event when case e is run with several values of R. As R increases fewer minimal cuts are touched and the associated overhead decreases. The value of $\overline{t}$ for given R is related to the number of minimal cuts (512 in the example). Then, from the figures shown in Table 3 we can conclude that, by taking an appropriate value for R, the overhead due to the bookkeepping of the failure distances will be in general small even if the model has tens of thousands of minimal cuts.

8 Conclusions

We have shown that the simulation of repairable fault-tolerant systems can be made very fast by exploiting the concept of failure distance. The failure distance biasing scheme proposed in this paper is more efficient than failure biasing in focussing the sampling to the paths with higher contributions and this results in reductions on the number of events required to achieve a given confidence interval which, as illustrated by the example presented, can be orders of magnitude. The efficiency of failure distance biasing in relation to failure biasing increases with the importance of coverage probabilities, which are poorly dealt with in failure biasing (they are sampled with very low probabilities due to the uncoverage factor, which is typically very small), and with the sparseness of combinations of k failed components in which the system is down for small values of k. As coverage failures are typically important and fault-tolerant systems are designed with good redundancy allocation so that down state with few components failed are sparse, failure distance biasing will be usually much more efficient than failure biasing.

Failure distance biasing is a very flexible scheme which can be adapted to a variety of scenarios and optimization of the parameters of the scheme is an important issue. We have proposed an optimization method which introduces negligeable overhead and typically takes the parameters to their optimal values after sampling a small number of paths. The proposed method can be applied to other biasing schemes of the same type.

We have also developed techniques for the computation of failure distances which introduce a small overhead even if the model has many minimal cuts. A limitation of our method is that it requires to find the minimal cuts of the model. Although theoretically the number of minimal cuts can be very large, in practice most fault-tolerant systems have a moderate number of minimal cuts so that the computational effort to find them is negligeable compared with the reduction in simulation times achieved by the proposed scheme over failure biasing. For instance, the 512 minimal cuts of the example presented in this paper were found in about 5 seconds of CPU time while simulation times were of the order of hours under failure baising and of the order of minutes with failure distance biasing.

Finally, it is likely that the failure distance concept can be exploited to improve current model pruning techniques giving error bounds [15, 16].

Acknowledgements

The author thanks Angel Calderón for implementing the algorithm to find the minimal cuts.

References

[1] E. E. Lewis and F. Böhm, "Monte Carlo Simulation of Markov Unreliability Models," *Nuclear Engineering and Design*, vol. 77, 1984, pp. 49–62.

[2] T. Zhuguo and E. E. Lewis, "Component Dependency Models in Markov Monte Carlo Simulation," *Reliability Engineering*, vol. 13, 1985, pp. 45–62.

[3] A. E. Conway and A. Goyal, "Monte Carlo Simulation of Computer Systems Availability/Reliability Models," *Proc. 17th Int. Symp. on Fault-Tolerant Computing (FTCS-17)*, 1987, pp. 230–235.

[4] R. M. Geist and M. K. Smotherman, "Ultrahigh Reliability Estimates through Simulation," *Proc. Ann. Reliability and Maintainability Symp.*, 1990, pp. 1–6.

[5] A. Goyal, P. Shahabuddin, P. Heidelbergeer, V. F. Nicola, and P. W. Glynn, "A Unified Framework for Simulating Markovian Models of Highly Dependable Systems," IBM Research Report RC 14772, Yorktown Heights, 1989.

[6] A. Goyal, P. Heidelberg and P. Shahabuddin, "Measure Specific Dynamic Importance Sampling for Availability Simulations," *Proc. Winter Simulation Conf.*, 1987, pp. 351–357.

[7] P. Shahabuddin, V. F. Nicola, P. Heidelberg, A. Goyal, and P. W. Glynn, "Variance Reduction in Mean Time to Failure Simulations," *Proc. Winter Simulation Conf.*, 1988, pp. 491–499.

[8] J. A. Carrasco, "Efficient Transient Simulation of Failure/Repair Markovian Models," Technical Report, UPC, 1990.

[9] R. E. Barlow and F. Proschan, *Statistical Theory of reliability and Life Testing. Probability Models*, McArdle Press, Silver Spring, 1981.

[10] J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*, Metheun, London, 1975.

[11] P. W. Glynn and J. L. Sanders, "Monte Carlo Optimization of Stochastic Systems: Two New Approaches," *ASME Computers in Engineering Conference*, 1986, pp. 219-223.

[12] J. A. Carrasco, "Failure Distance-based Simulation of Repairable Fault-Tolerant Systems," Technical Report, UPC, 1990.

[13] S. S. Rao, *Optimization, Theory and Applications*, Wiley Eastern, 1984.

[14] J. A. Carrasco and J. Figueras, "METFAC: Design and Implementation of a Software Tool for Modeling and Evaluation of Complex Fault-Tolerant Computing Systems," *Proc. 16th Int. Symp. on Fault-Tolerant Computing (FTCS-16)*, 1986, pp. 424–429.

[15] M. A. Boyd, M. Veeraraghavan, J. B. Dugan, and K. S. trivedi, "An Approach to Solving Large Reliability Models," *Proc. AIAA Components in Aerospace Conf.*, 1988, pp. 245–258.

[16] R. R. Muntz, E. de Souza e Silva, and A. Goyal, "Bounding Availability of Repairable Computer Systems," *IEEE Trans. on Computers*, vol. 38, no. 12, 1989, pp. 1714-1723.